IN THE CLAIMS:

Claims 1 and 10 are amended herein. Please cancel claims 19-27 without prejudice or disclaimer. All pending claims and their present status are produced below.

1. (Currently amended) A method for performing arithmetic in a memory to memory architecture in an embedded processor, the method comprising:

receiving a 32-bit fixed length instruction, the instruction specifying a source address in a memory using 11 bits, a source address in a register file using 5 bits, a destination address in the memory using 11 bits, and a mathematical operation to be performed using 5 bits; and

responsive to receiving the fixed length instruction:

accessing, from the source address in the memory, a first operand on which the mathematical operation is to be performed;

accessing, from the source address in the register file, a second operand on which the mathematical operation is to be performed;

performing the mathematical operation on the first operand and the second operand to obtain the result; and

storing the result in the destination address in the memory, wherein the destination address in the memory is different from the source address in the memory.

2. (Previously Presented) The method of claim 1, wherein the source address and the destination address in the memory correspond to 16 bit memory locations. 3.

(Previously Presented) The method of claim 1, wherein the source address and the destination address in the memory correspond to 8 bit memory locations.

4. (Previously Presented) The method of claim 1, further comprising the step of passing through a sign extender the first operand in response to the source address corresponding to a memory location of less than 32 bits.

5. (Previously Presented) The method of claim 4, further comprising truncating the result prior to storing in the destination address in the memory.

6. (Previously Presented) The method of claim 1, wherein the instruction specifies a size of the first operand, the size being one of 8 bits, 16 bits, and 32 bits.

7. (Previously Presented) The method of claim 6, wherein the size of the first operand is specified in an Operation Code within the 5 bits of the mathematical operation to be performed.

8. (Previously Presented) The method of claim 1, wherein the accessing the first operand further comprises, calculating the source address in the memory from data in the 11 bits of the 32-bit instruction specifying a source address in a memory.

9. (Previously Presented) The method of claim 8, wherein calculating comprises using one address mode of the group consisting of a Register + Immediate, a Register + Register Indirect, a Register + Immediate Auto-Increment, a Register Direct, and an Immediate addressing mode.

10. (Currently amended) An apparatus for performing arithmetic in a memory to memory architecture in an embedded processor, the apparatus comprising:

means for receiving a 32-bit fixed length instruction, the instruction specifying a source address in a memory using 11 bits, a source address in a register file using 5 bits, a destination address in the memory using 11 bits, and a mathematical operation to be performed using 5 bits;

means for accessing, from the source address in the memory, a first operand on which the mathematical operation is to be performed;

means for accessing, from the source address in the register file, a second operand on which the mathematical operation is to be performed;

means for performing the mathematical operation on the first operand and the second operand to obtain the result; and

means for storing the result in the destination address in the memory, wherein the destination address in the memory is different from the source address in the memory.

11. (Previously Presented) An embedded processor for providing connectivity in a communications system, comprising:

a 32-bit arithmetic-logic unit (ALU) comprising a first input, a second input, and an output, the ALU for performing an operation on a first 32-bit operand and a second 32-bit operand and for producing a 32-bit result, the operation specified in a 32-bit instruction fetched by the ALU;

a 32-bit register file for temporarily holding data, the 32-bit register file coupled to the first input of the ALU and communicatively coupled to the second input of the ALU for providing the first and the second 32-bit operand and coupled to the output of the ALU to receive the 32-bit result as an output from the ALU;

a memory device communicatively coupled with the second input of the ALU for providing the ALU input-data and communicatively coupled to the output of the ALU for receiving result-data, the memory device comprising a storage location of a size of less than 32 bits;

a sign extender coupled to the memory device for expanding the input-data from the memory to 32-bit data; and

a multiplexer comprising a first and a second input and an output, the first input coupled to the sign extender for receiving the expanded 32-bit data, the second input coupled to the 32-bit register, and the output coupled to the ALU, the multiplexer for selecting between the inputs the source for providing the first 32-bit operand to the ALU.

12. (Previously Presented) The embedded processor of claim 11, further comprising:

a truncator communicatively coupled to the ALU and the memory device, the truncator for converting the 32-bit result received from the ALU to a data unit of the size of the storage location in the memory device.

13. (Previously Presented) The embedded processor of claim 11, wherein the multiplexer further comprises a third input communicatively coupled to an immediate.

14. (Previously Presented) The embedded processor of claim 13, wherein the immediate is less than 32 bits and wherein the immediate is coupled to the sign expander for converting the immediate to an expanded 32-bit immediate, the sign expander further coupled to the third input of the multiplexer for communicating the expanded 32-bit immediate.

15. (Previously Presented) The embedded processor of claim 13, wherein the 32-bit instruction comprises a 5-bit OpCode for specifying the operation, an 11-bit source address for specifying a first source location for the first 32-bit operand, a 5-bit source address for specifying a second source location for the second 32-bit operand, and an 11-bit destination address for specifying a destination location to store the 32-bit result.

16. (Previously Presented) The embedded processor of claim 15, wherein the immediate comprises a 4-bit immediate and a 7-bit immediate, the embedded processor further comprising:

a general register file having no more than 32 data registers, the general register file comprising memory addressing information;

a second multiplexer comprising a first, a second, and a third input and an output, the first input coupled to the 7-bit immediate, the second input coupled to the 4-bit immediate, and the third input coupled to the general register file, the multiplexer for selecting between the inputs to provide an output;

an address register file having no more than 8 registers, the address register file comprising addressing information and an output;

an adder coupled to the output of the second multiplexer for receiving one of the 7-bit immediate, the 4-bit immediate, and the general register file, and the adder coupled to the address register for adding the output of the second multiplexer with the output of the address register and for providing a sum; and

a third multiplexer comprising a first input, a second input, and an output, the first input coupled to the address register file, the second input coupled to the adder for receiving the sum, the third multiplexer for choosing between the address register file output and the sum thereby providing the 11-bit source address of the 32-bit instruction according to an address mode.

17. (Previously Presented) The embedded processor of claim 16, wherein the address mode is one of the group consisting of a Register + Immediate, a Register + Register Indirect, a Register + Immediate Auto-Increment, a Register Direct, and an Immediate addressing mode.

18. (Previously Presented) The embedded processor of claim 11, wherein the 32-bit instruction comprises a 5-bit OpCode for specifying the operation, an 11-bit source address for specifying a first source location for the first 32-bit operand, a 5-bit source address for specifying a second source location for the second 32-bit operand, and an 11-bit destination address for specifying a destination location to store the 32-bit result.

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Cancelled)

26. (Cancelled)

27. (Cancelled)